

Bayesian Localization and Mapping Using GNSS SNR Measurements

Jason T. Isaacs¹, Andrew T. Irish¹, François Quitin², Upamanyu Madhow¹, and João P. Hespanha¹

Abstract—In urban areas, GNSS localization quality is often degraded due to signal blockage and multi-path reflections. When several GNSS signals are blocked by buildings, the remaining unblocked GNSS satellites are typically in a poor geometry for localization (nearly collinear along the street direction). Multi-path reflections result in pseudo range measurements that can be significantly longer than the line of sight path (true range) resulting in biased geolocation estimates. If a 3D map of the environment is available, one can address these problems by evaluating the likelihood of GNSS signal strength and location measurements given the map. We present two approaches based on this observation. The first is appropriate for cases when network connectivity may be unavailable or undesired and uses a particle filter framework that simultaneously improve both localization and the 3D map. This approach is shown via experiments to improve the map of a section of a university campus while simultaneously improving receiver localization. The second approach which may be more suitable for smartphone applications assumes that network connectivity is available and thus a software service running in the cloud performs the mapping and localization calculations. Early experiments demonstrate the potential of this approach to significantly improve geo-localization accuracy in urban areas.

I. INTRODUCTION

The widespread use of consumer electronics such as smartphones and tablets which are both network capable and Global Navigation Satellite Systems (GNSS) equipped has had an enormous impact on society. Real time location is critical to many mobile applications such as navigation, ride sharing, geo-fencing, and mobile coupons. However, in urban areas GNSS localization quality is often degraded due to signal blockage and multi-path reflections from buildings, trees, and other terrain [1]. In cluttered urban areas poor cross-street positioning accuracy results from some of the GNSS signals being blocked by buildings and leaving the remaining unblocked GNSS satellites in a poor geometry for localization. When signals reflect off of buildings but eventually reach the GNSS receiver the resulting pseudo range measurements can be significantly larger than the line-of-sight (LOS) path (true range), leading to large localization uncertainty.

In addition to geo-location coordinates, GNSS receivers also have the ability to record per-satellite identifier, azimuth,

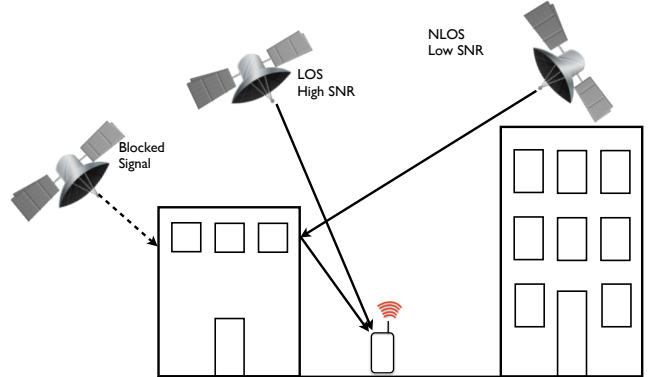


Fig. 1. Satellite SNR readings depend on LOS/NLOS path between receiver and satellite. NLOS paths are often characterized by low SNR.

elevation, and signal-to-noise ratio (SNR) information. A non-line-of-sight (NLOS) path, where the LOS path between a GNSS receiver and a particular satellite is occluded can be characterized by a statistically lower SNR when compared to the LOS SNR. See Figure 1 for an example scenario including both LOS and NLOS paths. By observing these SNR measurements, one can make inferences about the existence of NLOS/LOS channels (and thus obstacles/clear sky) in various directions relative to the geo-location of the GPS receiver. By fusing information from multiple receiver locations and multiple satellites, it becomes possible to determine the geo-location of obstacles. Based on this simple observation we propose to use a Bayesian framework to jointly build 3-dimensional maps of unknown environments and refine the receiver geo-location estimate.

In our prior work [2], we have shown that the posterior probability distribution of the map and receiver locations represents a factor graph, on which Loopy Belief Propagation (LBP) was used to efficiently estimate the probabilities of each cell being occupied or empty, along with the probability of the particles for each receiver location. By using the factor graph with Loop Belief Propagation approach we compute the full map posterior and geo-location estimates in one batch operation. In the work proposed here, a Bayes filter is used to estimate the posterior probability of occupancy for each grid cell individually, thus losing the ability to represent dependencies among neighboring cells. This tradeoff is made to allow for an algorithm that is suitable for use in real-time on a portable consumer electronics device such as a smartphone, tablet, or in car navigation system.

Contributions: We present two solutions to the simultaneous localization and mapping problem using only GNSS

*This work was supported by the Institute for Collaborative Biotechnologies through grant W911NF-09-0001 from the U.S. Army Research Office. The content of the information does not necessarily reflect the position or the policy of the Government, and no official endorsement should be inferred.

¹J.T. Isaacs, A.T. Irish, U. Madhow, and J.P. Hespanha are with Department of Electrical and Computer Engineering, University of California, Santa Barbara {jtisaacs, andrewirish, madhow, hespanha}@ece.ucsb.edu

²F. Quitin is with the School of Electrical and Electronic Engineering, Nanyang Technological University, fquitin@ntu.edu.sg

information. The first approach is similar to the grid-based FastSLAM algorithm [3] and can be summarized in the following steps. The map is modeled as a discrete set of binary state cells (occupancy grid), and the posterior distribution of both the map and the receiver's location is approximated by a set of weighted particles. At each time step, each prior particle is passed through a motion model to sample from the motion posterior. The importance weight of each particle is assigned based on the joint likelihood of each GNSS SNR and geo-location measurement returned by the GNSS receiver given the geo-location and map estimate of the particle. Next, each map particle is updated in a recursive manner using a log odds representation of occupancy and an inverse sensor model that maps SNR and satellite elevation measurements to probability of occupancy. Finally, particles are resampled with replacement based on the importance weights. These resampled particles form the priors for the next time step. The second approach is a hybrid solution that separates the mapping aspect of the problem from the real-time localization. The 3D map is periodically refined in a batch operation using techniques developed in [2] while the most recent map is used to perform particle filter based map matching. The map matching filter operates in a very similar fashion to the filter in the first approach without the need to maintain an estimate of the map state. This approach greatly reduces the computational complexity and thus may prove more attractive in practice.

Related Work: A technique which has been used to achieve significant localization improvement in cluttered urban areas where GNSS accuracy is often degraded is called *Shadow Matching* (SM) [1], [4]. Essentially, SM constrains the space of possible receiver locations by classifying signals as LOS/NLOS based on SNR readings and matching their points of reception to areas outside/inside the “shadows” of obstacles based on known 3D environment models. In [5], 3D maps are used to detect erroneous GNSS pseudoranges due to multipath reflections. These pseudoranges are then removed from the sensor fusion process resulting in improved geo-location accuracy. Implementation details of a real-time shadow matching smartphone positioning system are provided in [6], [7]. The shadow matching algorithm reduces cross-street position errors by around 70%. However, all shadow matching techniques rely on up-to-date 3D city models obtained from an external source which are not always available and can be expensive to obtain.

The problem of obtaining 3D environment models from GNSS signal strength measurements has received relatively little attention. Non-probabilistic heuristics based on ray tracing have been used to reconstruct environment maps after learning shadows of buildings from GNSS SNR measurements [8], [9]. In our prior work [2], a systematic Bayesian approach was used to simultaneously build 3D environment maps while correcting geo-location estimates of a large batch of GNSS receiver measurements. We believe that a probabilistic approach is more appropriate in general due to the large measurement uncertainty involved. To the best of our knowledge this was the first attempt to combine the

problems of localization improvement and 3D map building in the context of GNSS SNR measurements.

The application of Bayesian approaches to localization and mapping problems is quite common [10], often both the environment and sensor readings are modeled probabilistically. However, most Bayesian related approaches to SLAM are based on implicit or explicit measurements of distances to obstacles, using a variety of sensing methods such as lidar/radar [11], [12], mono/stereo camera [13], [14], and WiFi [15]. The GNSS SNR measurement model for a given satellite is quite different in that no range to obstacle information is available, only probabilistic LOS/NLOS information about the path to the satellite.

II. PROBLEM FORMULATION

We represent the unknown region with an *Occupancy Grid*. Formally, the occupancy grid is defined as a 3D grid of binary-valued “cells”, $m = \{m_i\}_{i=1}^L$, with $m_i \in \{0, 1\}$ where $m_i = 0$ denotes “empty” and $m_i = 1$ “occupied”. The space of possible GNSS receiver trajectories $x = \{x_t\}_{t=1}^T$ is represented using a set of weighted particles, so that individual positions are $x_t \in \{x_t^{[k]}\}_{k=1}^K$. The SLAM problem is then formulated as estimating the posterior distributions of each latent variable m and x given only the measurements available from commercially available GNSS receivers, namely geo-location coordinates of the receiver and per-satellite identifier, azimuth, elevation, and SNR.

A. SNR measurement model

The satellite SNR measurements, which are noisy and consist of T vector SNR readings, $z = \{z_t\}_{t=1}^T$, where $z_t = [z_{t,1}, \dots, z_{t,N_t}]$, and N_t is the number of satellites in view for the t th data sample. Associated with each SNR readings, are relative satellite elevations and azimuths $[\theta_{t,n}, \phi_{t,n}]$, which we consider noiseless. Under the assumption of a static world (where the map m does not change over time), the SNR measurements can be modeled as conditionally independent given the map and poses, yielding the following factorization

$$p(z|m, x) = \prod_{t,n} p(z_{t,n}|m, x_t). \quad (1)$$

Detailed statistical models exist for the narrowband Land to Mobile Satellite (LMS) channels of interest, such as those presented in [16], [17]. In previous work [2], we have proposed a simplified sensor model in which SNR readings are modeled differently based on LOS vs. NLOS. An SNR reading is LOS-distributed if all cells intersected by its associated receiver-satellite ray are empty; otherwise, it is NLOS-distributed. The SNR under LOS and NLOS hypotheses was modeled using Rician and log-normal distributions respectively. However, in this work we propose the following slightly more complicated yet similar empirically derived sensor model that also depends on satellite elevation,

$$p(z_{t,n}|m, x_t^k) = \begin{cases} f_{los}(z_{t,n}, \theta_{t,n}), & m_i = 0 \ \forall i \in \mathcal{M}(t, n, k) \\ f_{nlos}(z_{t,n}, \theta_{t,n}), & \text{otherwise} \end{cases} \quad (2)$$

where $\mathcal{M}(t, n, k)$ contains the indices of the cells intersected by the ray starting at particle $x_t^{[k]}$, in the direction of satellite n at time t . Example LOS/NLOS distributions are shown in Figure 2 for elevations of 15 and 60 degrees. Notice the NLOS distribution for the lower elevation satellite has a wide flat distribution to account for the fact that low elevation satellites typically provide noisier measurements. The detailed description and fitting of this model is beyond the scope of this document and left for future work.

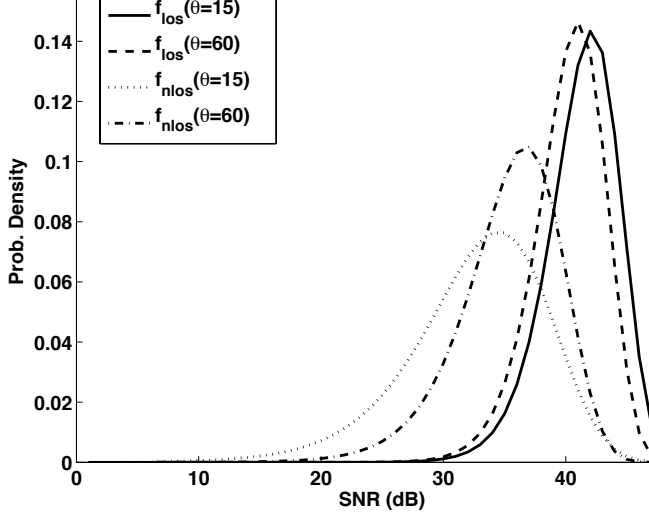


Fig. 2. The forward sensor model distributions for both LOS and NLOS satellite channels.

B. Geo-Location measurement model

The second type of information used are the receiver position estimates (GNSS fixes), which are noisy and modeled as independent Gaussian random variables

$$y_t = x_t + e_t, \quad e_t \sim \mathcal{N}(0, C_t). \quad (3)$$

As in Chapter 7 from [18], we estimate the error covariance matrix C_t using the formula for HDOP scaled by the uncertainty reported by the receiver. Let $\theta_t = [\theta_{t,1}, \dots, \theta_{t,N_t}]$ and $\phi_t = [\phi_{t,1}, \dots, \phi_{t,N_t}]$, and define the following $(3 \times N_t)$ matrix where each column is a unit vector,

$$H_t = \begin{bmatrix} \cos(\theta_t) .* \cos(\phi_t) \\ \cos(\theta_t) .* \sin(\phi_t) \\ \sin(\theta_t) \end{bmatrix} \quad (4)$$

where $.*$ indicates element by element multiplication. The DOP matrix is then given by,

$$C_t = \sigma_{URE}^2 (H_t H_t^T)^{-1} \quad (5)$$

where σ_{URE} represents the accuracy reported by the GNSS receiver.

III. GNSS PARTICLE FILTER SLAM

The first algorithm which we will refer to as GNSS Particle Filter SLAM is based on the grid-based FastSLAM algorithm from Chapter 13 in [19], but uses only passive measurements available from GNSS receivers. We assume that when the

algorithm is activated a prior map is available such as one built from crowd-sourced GNSS data as in our previous work [2]. A block diagram schematic showing data flow can be seen in Figure 3. Here a “master” map exists on a server and when network communication is available the relevant section of the map can be downloaded to the GNSS device. It may be that network communication becomes unavailable or undesirable due to data usage concerns. In this case, the GNSS device can use the the GNSS Particle Filter SLAM algorithm to both update the prior map and geo-location estimate.

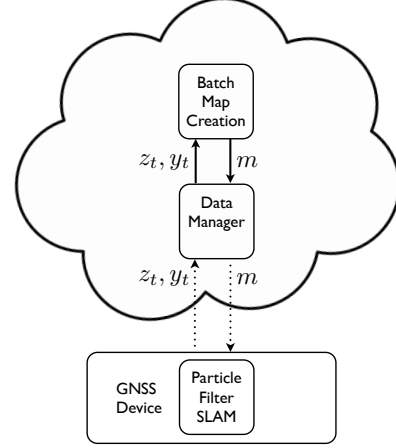


Fig. 3. The block diagram description of data flow between the GNSS Particle Filter SLAM algorithm and the server containing the “master” map.

The GNSS Particle Filter SLAM algorithm is summarized along with function interface definitions for each of the required functions in Algorithm 1. Here each particle contains an estimate of both geo-location and the map, therefore M copies of the map must be maintained. The k th copy of the map at time t is denoted as $m_t^{[k]}$ and contains L cells such that the i th cell of the k th map is denoted as $m_{t,i}^{[k]}$.

Algorithm 1 GNSS Particle Filter SLAM($\mathcal{X}_{t-1}, z_t, y_t$):

- 1: $\bar{\mathcal{X}}_t = \mathcal{X}_t = \emptyset$
 - 2: **for** $k = 1$ to M **do**
 - 3: $x_t^{[k]} = \text{sample-motion-model}(x_{t-1}^{[k]})$
 - 4: $w_t^{[k]} = \text{measurement-model-map}(z_t, y_t, x_t^{[k]}, m_{t-1}^{[k]})$
 - 5: $m_t^{[k]} = \text{update-map}(z_t, y_t, x_t^{[k]}, m_{t-1}^{[k]})$
 - 6: $\bar{\mathcal{X}}_t = \bar{\mathcal{X}}_t + \langle x_t^{[k]}, m_t^{[k]}, w_t^{[k]} \rangle$
 - 7: **end for**
 - 8: **for** $k = 1$ to M **do**
 - 9: draw i with probability $\propto w_t^{[i]}$
 - 10: add $\langle x_t^{[i]}, m_t^{[i]} \rangle$ to \mathcal{X}_t
 - 11: **end for**
 - 12: **return** \mathcal{X}_t
-

- **sample-motion-model**($x_{t-1}^{[k]}$)

To represent the motion model for the receiver we have

chosen the following linear model,

$$\begin{aligned} x_t^{[k]} &= f(x_{t-1}^{[k]}, d_t) \\ &= \begin{bmatrix} 1 & 0 & 1 & 0 \\ 0 & 1 & 0 & 1 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} x_{t-1}^{[k]} + d_t, \end{aligned} \quad (6)$$

where d represents additive process noise. Note a more appropriate motion model $f(\cdot)$ could be chosen if more information were known about the transportation of the receiver (i.e. if it were in a vehicle).

- **measurement-model-map**($z_t, y_t, x_t^{[k]}, m_{t-1}^{[k]}$)

Next we need a function that computes $w_t^{[k]}$, the likelihood of the measurements z_t and y_t given the pose $x_t^{[k]}$ represented by the k -th particle and given the map $m_{t-1}^{[k]}$ computed at the previous measurement index. From (3) we see that

$$p(y_t | x_t^{[k]}) = \mathcal{N}(y_t; x_t^{[k]}, C_t) \quad (7)$$

The measurement model for map matching and SNR is a bit more complicated but can be written as,

$$\begin{aligned} p(z_t | x_t^{[k]}) &= \prod_{j=1}^{N_t} \left(\prod_{i \in \mathcal{M}(t-1, j, k)} 1 - p(m_{t-1, i}^{[k]}) \right) f_{los}(z_{t, j}, \theta_{t, j}) \\ &+ \left(1 - \prod_{i \in \mathcal{M}(t-1, j, k)} 1 - p(m_{t-1, i}^{[k]}) \right) f_{nlos}(z_{t, j}, \theta_{t, j}) \end{aligned} \quad (8)$$

Combining the two,

$$w_t^{[k]} = p(y_t | x_t^{[k]}) p(z_t | x_t^{[k]}) \quad (9)$$

- **update-map**($z_t, x_t^{[k]}, m_{t-1}^{[k]}$)

Additionally, we need a function that updates the occupancy grid map, given the current pose $x_t^{[k]}$ of the k -th particle, the measurements z_t and y_t , and the map $m_{t-1}^{[k]}$ computed at the previous measurement index. The update-map algorithm summarized in Algorithm 2 is a standard binary Bayes filter with a log-odds representation of occupancy.

Algorithm 2 update-map($z_t, y_t, x_t^{[k]}, m_{t-1}^{[k]}$):

```

1: for  $i = 1$  to  $L$  do
2:    $l_{t-1, i} = \log(m_{t-1, i}^{[k]} / (1 - m_{t-1, i}^{[k]}))$ 
3:   for  $n = 1$  to  $N_t$  do
4:     if  $i \in \mathcal{M}(t, n, k)$  then
5:        $l_{t, i} = l_{t-1, i} + \text{inverse-sensor-model}(z_{t, n}, \theta_{t, n}) - l_0$ 
6:     else
7:        $l_{t, i} = l_{t-1, i}$ 
8:     end if
9:   end for
10:   $m_{t, i}^{[k]} = 1 - 1 / (1 + \exp(l_{t, i}))$ 
11: end for
12: return  $m_t^{[k]}$ 
```

The last remaining function to describe is

- **inverse-sensor-model**(z_t, θ_t)

Prior to conversion to log-odds form, the inverse sensor model is,

$$m_{t, i}^{[k]} = \alpha - \frac{\arctan\left((z_{t, n} - \bar{z}) \left(\frac{1 - \cos(\theta_{t, n})}{\lambda}\right)\right)}{\pi}, \quad (10)$$

and can be seen in Figure 4. The tuning parameters α , λ , and \bar{z} can be used to adjust how aggressively the map will be altered by a single measurement. The values of $\{\alpha = 0.5, \lambda = 5, \bar{z} = 30\}$ were used in Figure 4 and were chosen to be very conservative with a goal of doing no harm to the existing map. For instance low elevation satellites which can have very noisy SNR readings have little bearing on the map regardless of the SNR value. Additionally, higher elevation satellites have a significant influence only if the SNR value deviates far from the center value.

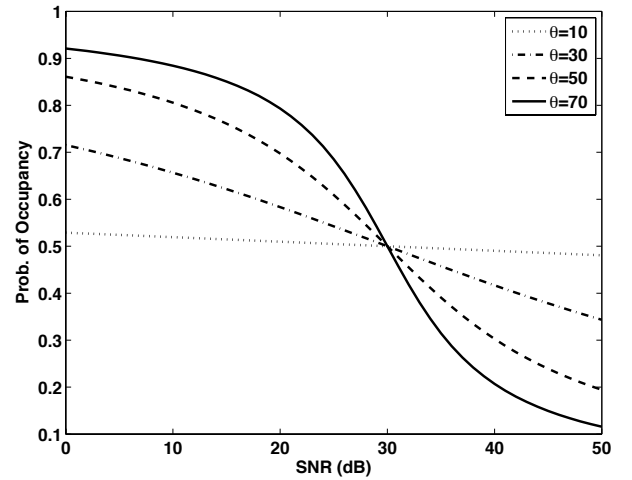


Fig. 4. The inverse sensor model that maps GNSS SNR and elevation to probability of occupancy.

After performing the importance based resampling in the second for-loop of the algorithm we are left with a set of particles \mathcal{X}_t which contains M geo-location particles and M occupancy grid maps. If the goal is to use the SLAM algorithm to improve localization, then geo-location particles can be fused to report a single estimate of geo-location x_t .

IV. GNSS PARTICLE FILTER MAP MATCHING

The second algorithm which we will refer to as GNSS Particle Filter Map Matching is similar to Algorithm 1 without the map update step thus eliminating the need to maintain M copies of the map. It is our vision that this algorithm would run on the cloud, but it could conceivably run on the GNSS device provided that network availability is sufficient to maintain downloads of them most up to date map from the server. A block diagram schematic showing data flow can be seen in Figure 5. Here a “master” map is periodically built in a batch process from crowd sourced GNSS data as in our previous work [2]. The relevant section of the “master” map can be passed to the GNSS Particle Filter Map Matching algorithm by the Data Manager as needed.

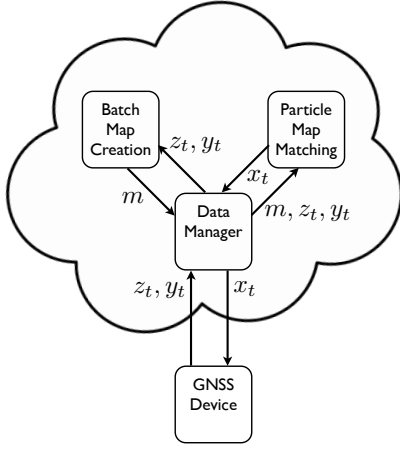


Fig. 5. The block diagram description of data flow between the GNSS Particle Filter Map Matching algorithm and the server containing the “master” map.

For completeness, the GNSS Particle Filter Map Matching algorithm is summarized in Algorithm 3 along with function interface definitions for each of the required functions.

Algorithm 3 GNSS Map Matching($\mathcal{X}_{t-1}, m, z_t, y_t$):

- 1: $\bar{\mathcal{X}}_t = \mathcal{X}_t = \emptyset$
 - 2: **for** $k = 1$ to M **do**
 - 3: $x_t^{[k]} = \text{sample-motion-model}(x_{t-1}^{[k]})$
 - 4: $w_t^{[k]} = \text{measurement-model-map}(z_t, y_t, x_t^{[k]}, m)$
 - 5: $\bar{\mathcal{X}}_t = \bar{\mathcal{X}}_t + \langle x_t^{[k]}, w_t^{[k]} \rangle$
 - 6: **end for**
 - 7: **for** $k = 1$ to M **do**
 - 8: draw i with probability $\propto w_t^{[i]}$
 - 9: add $\langle x_t^{[i]}, w_t^{[i]} \rangle$ to \mathcal{X}_t
 - 10: **end for**
 - 11: **return** \mathcal{X}_t
-

The functions **sample-motion-model**($x_{t-1}^{[k]}$) and **measurement-model-map**($z_t, y_t, x_t^{[k]}, m$) are identical to (6) and (7)-(9) with the exception that (8) uses the “master” map instead of a particle estimate of the map.

V. EXPERIMENTAL RESULTS

To verify the efficacy of the proposed algorithms we used a Samsung Galaxy Tablet 2.0 running the Android operating system to collect GPS/Glonass information along known paths on the eastern corner of the University of California, Santa Barbara campus (see Figure 7). This recorded data set was then fed as inputs to each algorithm proposed in this work. The map in Figure 6 was generated using information from Open Street Maps and algorithms presented in [2], and was treated as the “master” map in these experiments. This map uses cells of dimension $3\text{ m} \times 3\text{ m} \times 3\text{ m}$ has a max height of 24 m .

A. GNSS Particle Filter SLAM

The first goal of the GNSS Particle Filter SLAM algorithm is to improve localization. A data-set corresponding to a

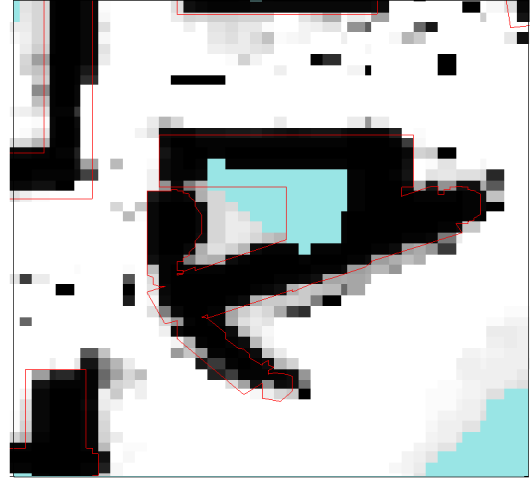


Fig. 6. The horizontal slice (3-6 meter height) of the prior map of UCSB campus used in experiments. Cells with high/low probability of occupancy are colored with black/white. The red lines indicate building boundaries according to Open Street Maps.

known path by the receiver was recorded for analysis, and the resulting geo-location estimates can be seen in Figure 7. The known path of the receiver is shown as a dashed black line, and the receiver latitude/longitude fixes and corresponding uncertainty ellipses are shown in blue. The improved position estimate and an ellipse corresponding to the sample covariance of the particles are shown in cyan. Of particular interest are the points in the north-west corner of the building where the original fix has errors of several meters in the direction of the building. The proposed algorithm pushes the particles away from the building and back on the sidewalk near the true path. Additionally, the resulting map would have underestimated the occupancy of the cells in this area without this position correction.



Fig. 7. The resulting geo-location improvement from Algorithm 1. The blue ellipses represent geo-location reported by the GNSS receiver. The cyan ellipses represent the geo-location estimates resulting from the Algorithm 1. The black dashed line represents ground truth of the GNSS receiver trajectory.

The second goal of the GNSS Particle Filter SLAM algorithm is to maintain and improve the quality of the initial 3D city map. The resulting map from running Algorithm 1 on

the sampled data-set was compared to the original “master” map from Figure 6. The horizontal slice corresponding to 3 – 6 m and 6 – 9 m heights can be seen in Figures 8 and 9 respectively. The color of each cell indicates the difference between the probability of occupancy before and after running Algorithm 1. Dark cells indicate increasing probability of occupancy. The red lines indicate building boundaries according to Open Street Maps. Notice that most of the cells are unchanged, but most of the changes to the map resulted in increasing the probability of occupancy for cells inside the red contour.

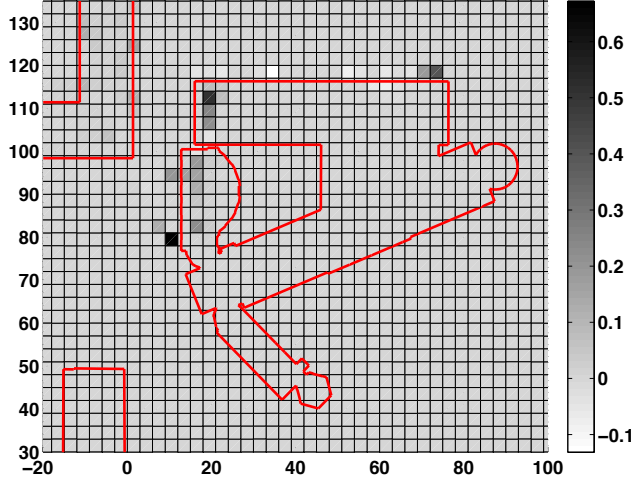


Fig. 8. After applying Algorithm 1 the resulting changes to the horizontal slice (3-6 meter height) of the map.

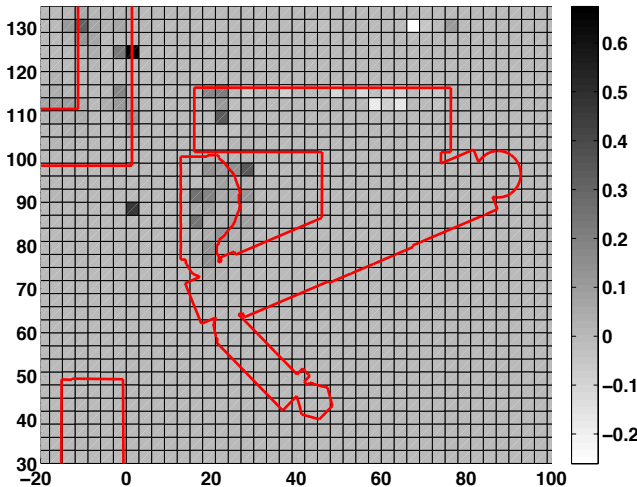


Fig. 9. After applying Algorithm 1 the resulting changes to the horizontal slice (6-9 meter height) of the map.

B. GNSS Particle Filter Map Matching

The same data-set from above was used to evaluate the Particle Filter Map Matching algorithm, and the resulting geo-location estimates can be seen in Figure 10. The known path of the receiver is shown as a dashed black line,

and the receiver latitude/longitude fixes and corresponding uncertainty ellipses are shown in blue. The improved position estimate and an ellipse corresponding to the sample covariance of the particles are shown in cyan. The geo-location improvements are very similar to Figure 7, as was expected since the “master” map from Figure 6 was accurate to begin with. If the primary goal is geo-localization improvement and a high quality map exists, then Algorithm 3 performs well without the memory overhead of Algorithm 1.



Fig. 10. The resulting geo-location improvement from Algorithm 3. The blue ellipses represent geo-location reported by the GNSS receiver. The cyan ellipses represent the geo-location resulting from using the map matching algorithm. The black dashed line represents ground truth of the GNSS receiver trajectory.

VI. CONCLUSION

Two approaches were presented to help alleviate large geo-location errors in urban environments due to GNSS blocked signals and multi-path reflections. Both approaches assumed the existence of a prior 3D environment map and uses the map to infer when GNSS signals are LOS or NLOS. This information is fused with the geo-location estimate provided by the GNSS receiver and an appropriate motion model in a particle-based Bayes filter framework. The first approach seeks to simultaneously improve both localization and the 3D map. This approach may be appropriate when network connectivity is unavailable or undesired but comes with the caveat of requiring enough memory to maintain a large number of map estimates. Initial experiments, conducted on UCSB campus, demonstrate the ability of this algorithm to both improve the map while simultaneously improving receiver localization. The second approach relies on a software service running in the cloud to perform the mapping and localization calculations. Here a single “master” map is stored (and periodically updated) on a server and a particle-based Bayes filter is used to perform map matching. Early experiments, conducted on UCSB campus, show a comparable geo-location improvement to the first approach without the memory requirements necessary to maintain many map estimates. Ongoing work involves implementing the proposed approaches in a cloud computing framework, developing a mobile application, and improving the methods used to build and maintain the “master” maps.

ACKNOWLEDGEMENT

The authors would also like to acknowledge Adam Ehrlich for developing the Android application used to log GPS data.

REFERENCES

- [1] P. Groves, "Shadow matching: A new GNSS positioning technique for urban canyons," *Journal of Navigation*, vol. 64, no. 3, pp. 417–430, 2011.
- [2] A. T. Irish, J. T. Isaacs, F. Qutin, J. P. Hespanha, and U. Madhwa, "Belief propagation based localization and mapping using sparsely sampled GNSS SNR measurements," in *Proc. of the International Conference on Robotics and Automation*, 2014.
- [3] M. Montemerlo, S. Thrun, D. Koller, and B. Wegbreit, "FastSLAM: a factored solution to the simultaneous localization and mapping problem," in *AAAI/IAAI*, 2002, pp. 593–598.
- [4] B. Ben-Moshe, E. Elkin, H. Levi, and A. Weissman, "Improving accuracy of GNSS devices in urban canyons," in *Proc. of the 23rd Canadian Conference on Computational Geometry*, 2011.
- [5] M. Obst, S. Bauer, and G. Wanielik, "Urban multipath detection and mitigation with dynamic 3D maps for reliable land vehicle localization," in *Proc. of the Position Location and Navigation Symposium*, 2012, pp. 685–691.
- [6] L. Wang, P. D. Groves, and M. K. Ziebart, "Urban positioning on a smartphone: Real-time shadow matching using gnss and 3d city models," in *Proc. ION GNSS*, 2013.
- [7] —, "Shadow matching: Improving smartphone GNSS positioning in urban environments," in *Proc. of the China Satellite Navigation Conference*, 2013, pp. 613–621.
- [8] K. Kim, J. Summet, T. Starner, D. Ashbrook, M. Kapade, and I. Essa, "Localization and 3D reconstruction of urban scenes using GPS," in *Proc. of the IEEE International Symposium on Wearable Computers*, 2008, pp. 11–14.
- [9] A. Weissman, B. Ben-Moshe, H. Levi, and R. Yozevitch, "2.5D mapping using GNSS signal analysis," in *Proc. of the Workshop on Positioning Navigation and Communication*, 2013, pp. 1–6.
- [10] S. Thrun, "Robotic mapping: A survey," *Exploring Artificial Intelligence in the New Millennium*, vol. 1, pp. 1–35, 2003.
- [11] S. Thrun, W. Burgard, and D. Fox, "A real-time algorithm for mobile robot mapping with applications to multi-robot and 3D mapping," in *Proc. of the International Conference on Robotics and Automation*, vol. 1, 2000, pp. 321–328.
- [12] M. W. M. G. Dissanayake, P. Newman, S. Clark, H. Durrant-Whyte, and M. Csorba, "A solution to the simultaneous localization and map building (SLAM) problem," *IEEE Trans. on Robotics and Automation*, vol. 17, no. 3, pp. 229–241, 2001.
- [13] A. J. Davison, I. D. Reid, N. D. Molton, and O. Stasse, "MonoSLAM: Real-time single camera SLAM," *IEEE Trans. on Pattern Analysis and Machine Intelligence*, vol. 29, no. 6, pp. 1052–1067, 2007.
- [14] P. Elinas, R. Sim, and J. J. Little, "spl sigma/SLAM: Stereo vision SLAM using the Rao-Blackwellised particle filter and a novel mixture proposal distribution," in *Proc. of the International Conference on Robotics and Automation*, 2006, pp. 1564–1570.
- [15] B. Ferris, D. Fox, and N. Lawrence, "WiFi-SLAM using Gaussian process latent variable models," in *Proc. of the 20th International Joint Conference on Artificial Intelligence*, 2007, pp. 2480–2485.
- [16] A. Abdi, W. C. Lau, M.-S. Alouini, and M. Kaveh, "A new simple model for land mobile satellite channels: first-and second-order statistics," *IEEE Trans. on Wireless Communications*, vol. 2, no. 3, pp. 519–528, 2003.
- [17] C. Loo, "A statistical model for a land mobile satellite link," *IEEE Trans. on Vehicular Technology*, vol. 34, no. 3, pp. 122–127, 1985.
- [18] E. D. Kaplan and C. J. Hegarty, *Understanding GPS: Principles and Applications*, 2nd ed. Artech House, 2005.
- [19] S. Thrun, W. Burgard, and D. Fox, *Probabilistic robotics*. MIT Press, 2005.